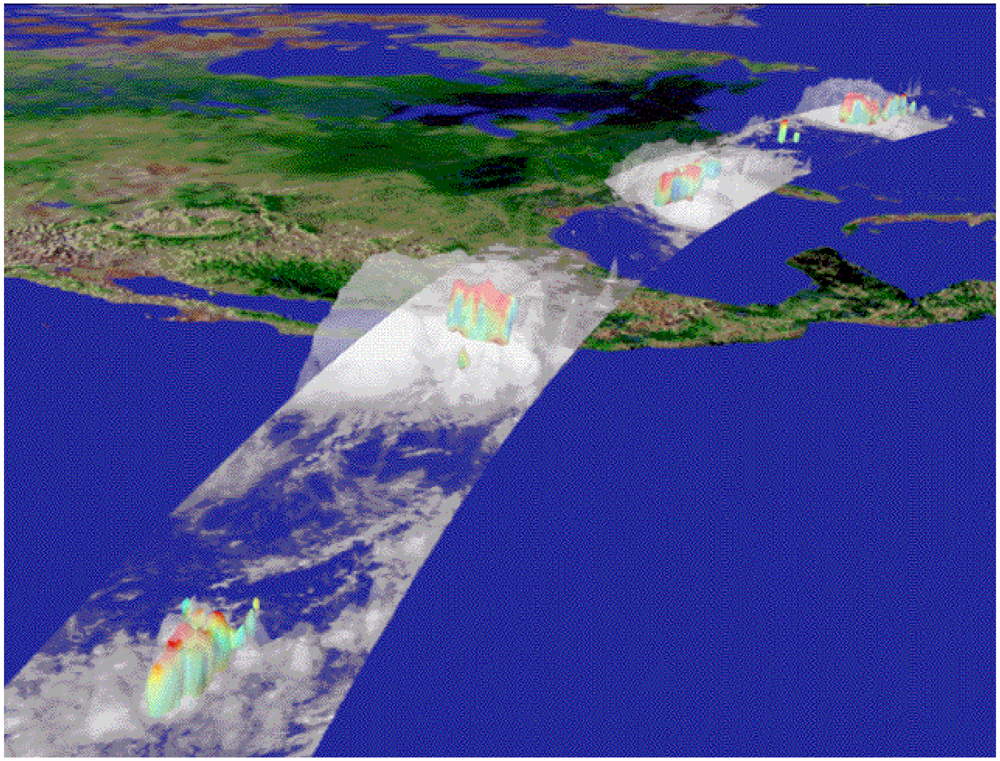
An On-Board Processor for a Spaceborne Doppler Precipitation Radar: Requirements and Preliminary Design

S. Durden, M. A. Fischman, S. Tanelli, R. A. Johnson, A. Chu

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109

Overview of Spaceborne Precipitation Radar



Tropical Storm Howard, Hurricane Isis, Hurricane Earl and Hurricane Danielle all line up under TRMM, passing over the scene from West to East on Sept. 2, 1998. (Courtesy of NASA/NASDA TRMM Project).

- TRMM Radar characteristics
  - 14-GHz
  - 2-m antenna
- TRMM Radar measures 3-D reflectivity structure
- TRMM does not measure atmospheric motion
  - antenna is too small
- Measurements of vertical motion are needed for
  - latent heating
  - improved rain classification
- Future generations of spaceborne precipitation radar are likely to have Doppler capability to meet this need

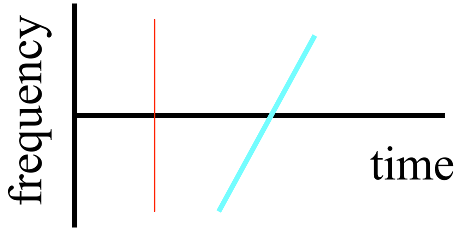
Overview of Doppler Processing

- Most weather radars use the pulse-pair method to estimate the Doppler velocity
  - Form lag-1 complex auto-correlation R(1) at each range bin
  - Compute argument of R(1) and convert to velocity
- This technique can work well for a spaceborne radar
  - only if the rain within the beam is fairly uniform
- For substantial inhomogeneity within the beam, the Doppler estimate is biased
  - For example a point target located forward of nadir would have a Doppler shift due to platform motion, even if the target itself is stationary
- A Doppler accuracy of 1 m/s is desirable; simulations with realistic reflectivity distributions indicate errors of up to several m/s
- An evaluation of these effects has been published in S. Tanelli, E. Im, S. L. Durden, L. Facheris, and D. Guili, “The effects of nonuniform beam filling on vertical rainfall velocity measurements with spaceborne Doppler radar," J. Atmos. Oceanic Technol., vol. 19, pp. 1019-1034, July 2002

Combined Frequency-Time (CFT) Method

- In pulse-pair one is effectively integrating the Doppler time-frequency spectrum along lines of constant time (to compute the Doppler centroid).
- In developing the CFT it was noted that a point target traces a trajectory in time-frequency space as it passes through the antenna beam
- By processing data along these trajectories, we deal with targets that have passed all the way through the beam
- Simulations with CFT on aircraft-measured reflectivity fields have demonstrated accuracy for convective rainfall is improved to about 1 m/s
- This technique is described in S. Tanelli, E. Im, S. L. Durden, L. Facheris, D. Guili, E. A. Smith, “Rainfall Doppler velocity measurements from spaceborne radar: Overcoming NUBF effects," J. Atmos. Oceanic Technol., vol. 21, pp. 27-44, Jan 2004

At right, red line is conventional integration axis for computing Doppler centroid. Blue line is axis corresponding to target motion through beam.

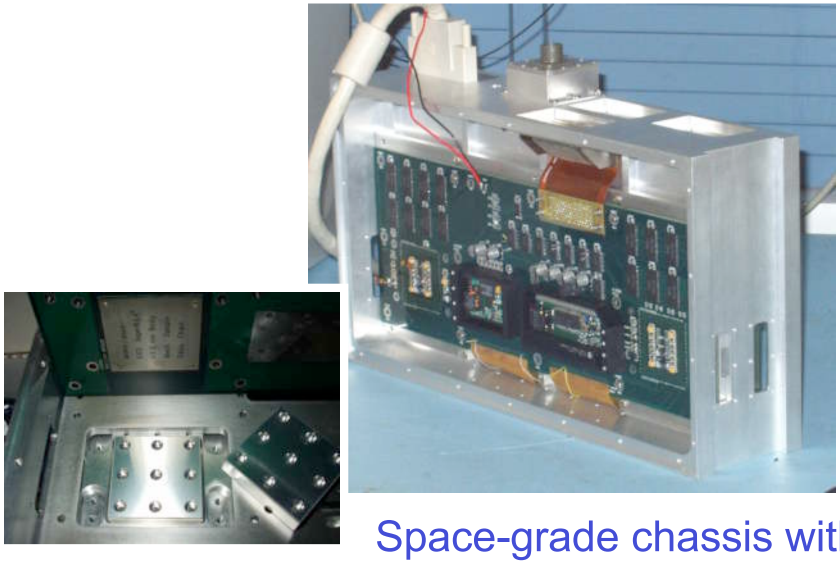


Processor Overview

- On-board Doppler spectral processing is needed to reduce data volume prior to down-linking; CFT can be performed on down-linked spectra
- Doppler processor will accept as input a contiguous block of pulses (2-d data, 64 pulses by 480 range bins, based on 2nd generation precipitation radar)
- At each range bin, the following processing steps will be performed:
  - Windowing: the complex input samples at a given range bin are multiplied by a windowing function
  - FFT: a fixed point Fast Fourier Transform of each block of complex samples is calculated
  - Magnitude squared: the magnitude squared of the FFT complex output is computed; this provides the Doppler power spectrum at each range bin
  - Averaging in range: the spectra from several range bins are averaged to reduce the effects of fading, with added benefit of reducing the data volume/rate
- Processor will be implemented using the Xilinx V-1000 part (FPGA)
  - compatible with existing hardware
  - has equivalent radiation tolerant part for space (demonstrated on MER)

Space-Grade Radar Processor Hardware

- Design for space environment
  - All S-class equivalent parts (Virtex & Actel FPGAs, SRAM, buffers)
  - Configuration memory scrubbing for single-event upset recovery
  - Chassis design: Conductive cooling of Data Processor FPGAs



Space-grade chassis with assembled printed circuit boards.

Benchtop testing and installation in thermal chamber.

Experimental setup

- Designed benchtop PC data acquisition and control system
  - LabView GUI software + 32-bit digital I/O interface
- Testing with a target simulator (chirp waveform generator)
  - Can run entire adaptive scan cycles in real time
  - Processor operation confirmed in thermal chamber tests over –20 to +70 °C

Processor Requirements

- **Input Data:** 64 complex pulses, each pulse has 480 range bins. Each complex number is two 16-bit integers (real and imaginary, or I and Q). The integers are signed, with 2’s complement representation.
- **Doppler Accuracy**: spectra which can be processed with the CFT to get accuracy in Doppler centroid of 1 m/s
- **Dynamic Range:** 60 dB to accommodate variation within a single spectrum and variability within the 14 km data window
- **Total Processing Time**: 11 ms to maintain real-time operation (64 pulses at 6000 Hz pulse repetition frequency)
- **Magnitude-Squared Operation**
  - 17-bit I/Q input data is expanded to 33 bits in this operation
  - Result is truncated to 29 bits, with selection of 29 out of 33 to be programmable by setting a 3-bit register
- **Averaging Operation**
  - Spectra are summed over 8 range bins
  - This expands the 29 bit values to 32 bits

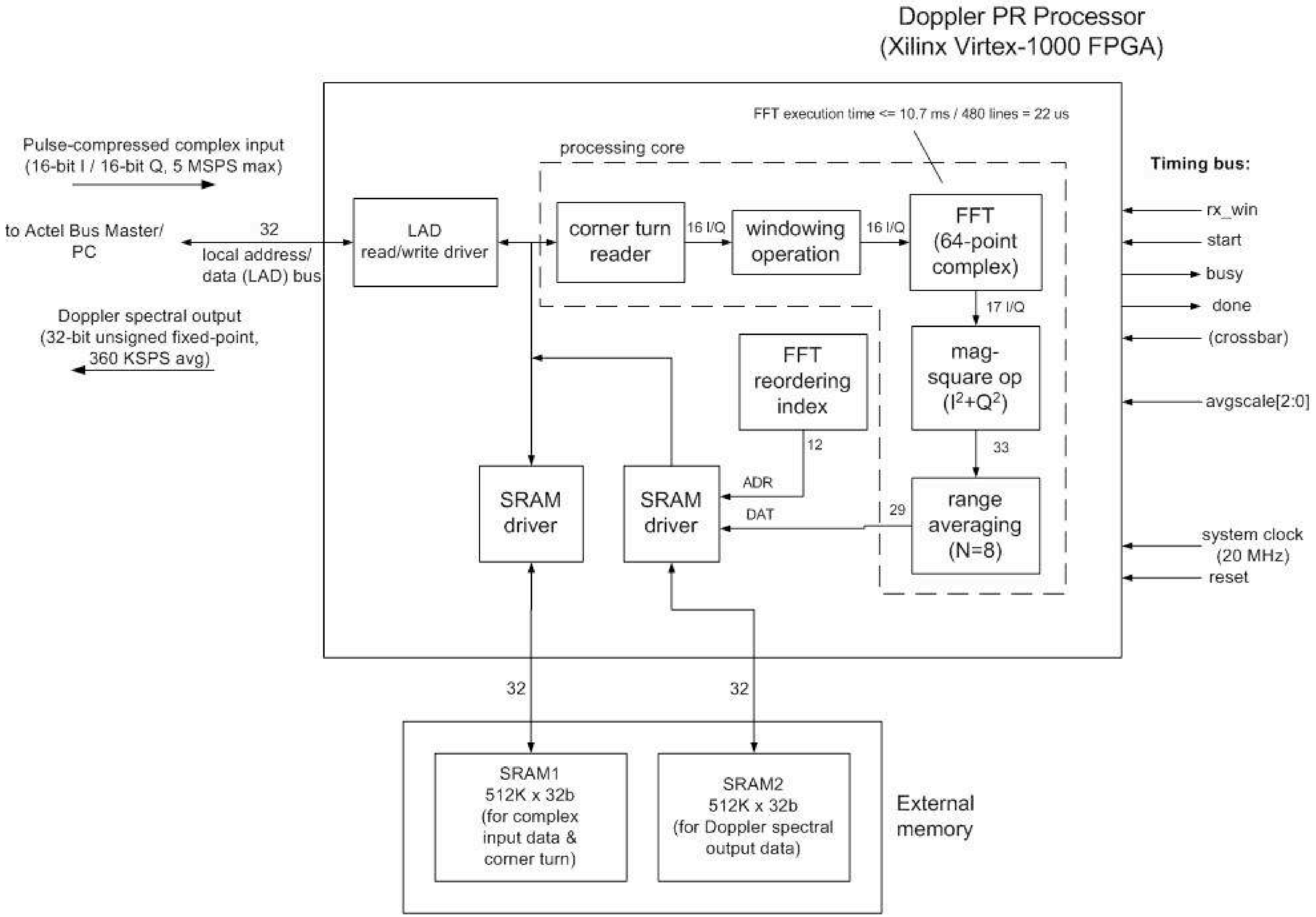
FFT Requirements

- Execution time
  - 22 microseconds, based on sequential FFT processing; 11 ms total divided by 480 range bins.
  - assumes input and output are pipelined to allow full time for FFT; other operations are assumed to required insignificant time.
- Precision and scaling
  - The SNR for uniformly distributed input has been previously solved analytically; it is 6.02b-10log(4N), b is the number of bits, N is FFT length
    - has been verified approximately using a 16-bit integer C-code simulation
  - This result indicates that required dynamic range can be achieved using 15-bits plus sign bit.
  - Weather radar echoes have normally distributed real and imaginary components
  - simulations with these statistics indicate that butterflies can overflow in some situations, so 17-bit arithmetic is needed in the FFT (16 mag bits and sign bit).

System Design

- All modules are designed to run synchronously off 20 MHz system clock
- Input signal consists of
  - 16-bit I / 16-bit Q complex baseband range bin samples
  - sample rate of 5 Msample/s (160 Mbit/s peak rate)
  - via a bidirectional local address and data (LAD) bus
- Over the 11 ms processing interval, the received I/Q data from 64 radar pulses are stored in external static RAM in
  - rows (representing the 480 range altitude bins) and in
  - columns (representing the 64 successive radar pulses along-track)
- After processing interval is captured in memory, an SRAM driver circuit reads the data back into the processor row-by-row in a “corner turn” fashion
  - FFT processing can be performed on range bin samples at each altitude
  - Crossbar memory allows simultaneous processing of previous interval while receiving radar echoes from current interval
- After processing, the averaged Doppler spectral data is written to a second bank of external SRAM
  - lower 6 offset address bits reversed
- Output rate on LAD bus 11.5 Mbit/s

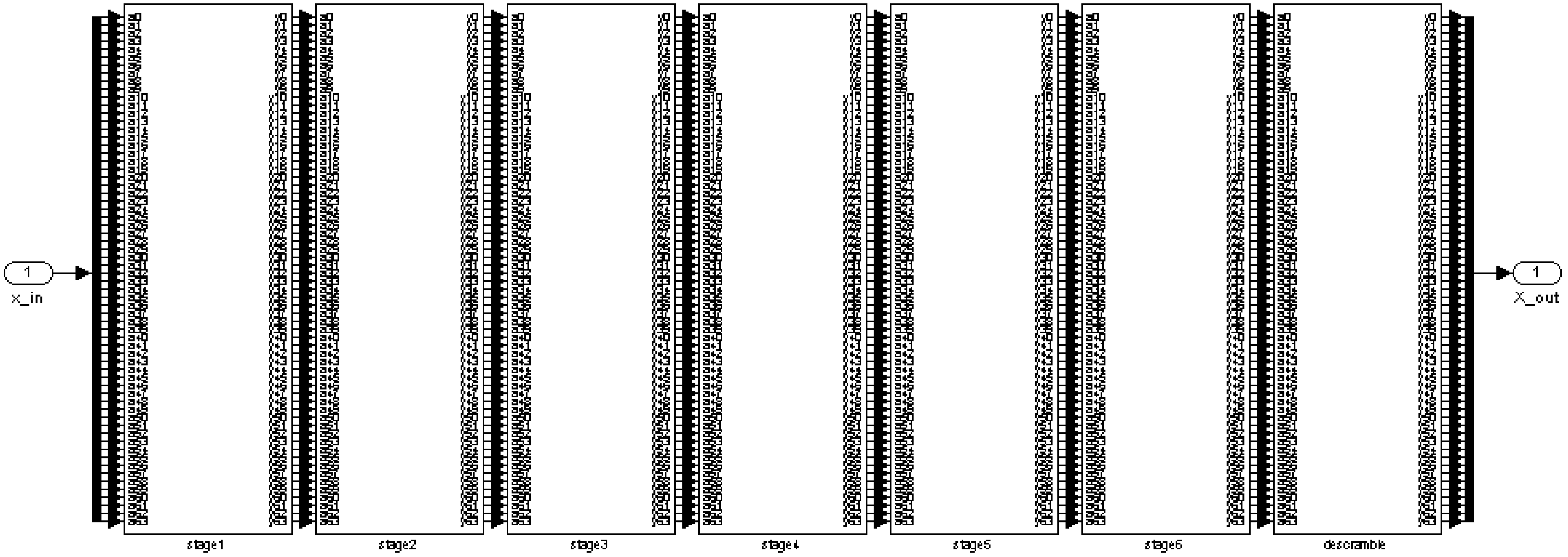
Core Functions and Data Interface



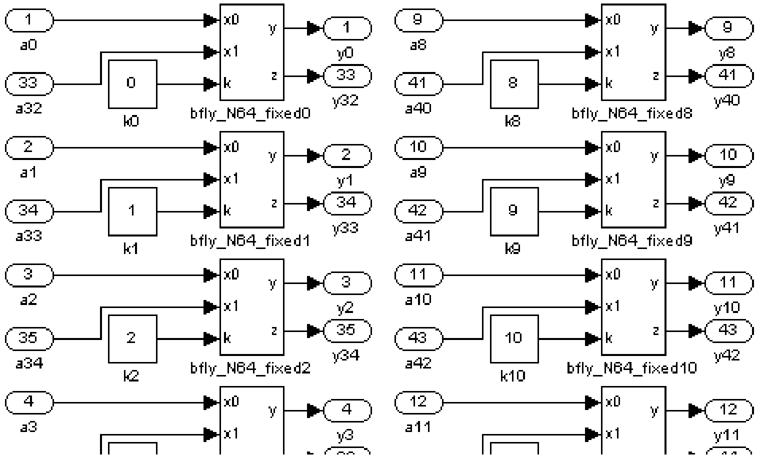
Bit-True Simulations

- To analyze how the Doppler processing algorithm could be implemented in FPGA hardware, a bit-true model of the processor was developed in Simulink using the “Fixed-Point Blockset” library
  - includes fixed-point adders and multipliers which emulate the actual effects of DSP hardware such as numeric overflows or quantization
- The key component in the fixed-point Doppler processor design is the N=64 point, radix-2 FFT
- A decimation-in-frequency approach was chosen; it consists of $\log_2 N = 6$ stages, each stage containing $N/2 = 32$ adder/multiplier butterfly operations
  - in all 192 butterfly operations are required to complete one FFT operation
- The butterfly architecture uses 16-bit I/Q resolution on the twiddle factor and 17-bit resolution on the complex input and output data pair
  - prevents overflows during the complex phasor rotation (multiplication) by the twiddle factor
  - 1 butterfly has 2 complex additions + 1 complex multiplication = 10 real operations

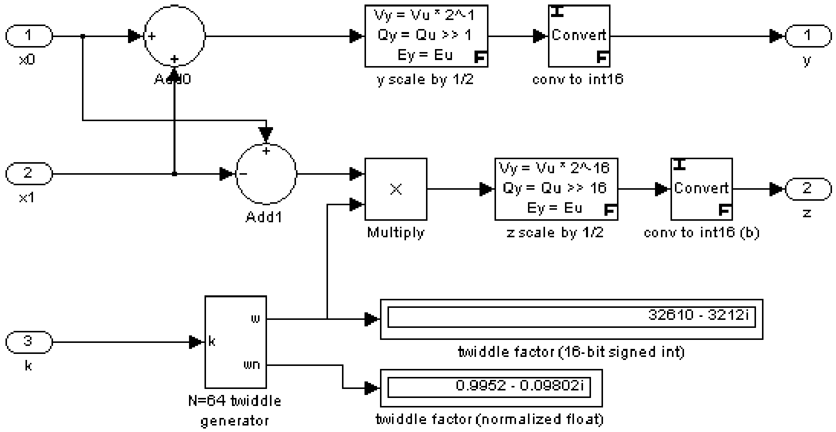
FFT Fixed-Point Algorithm Design (Simulink)



Top level (64-point / 6 stage FFT)



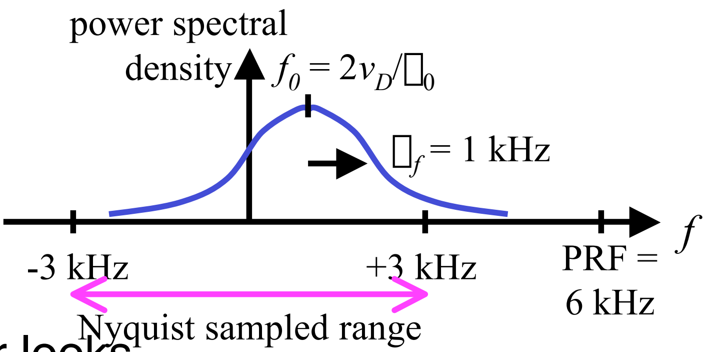
FFT stage



Fixed-point butterfly w/ twiddle factor memory

Simulation Method

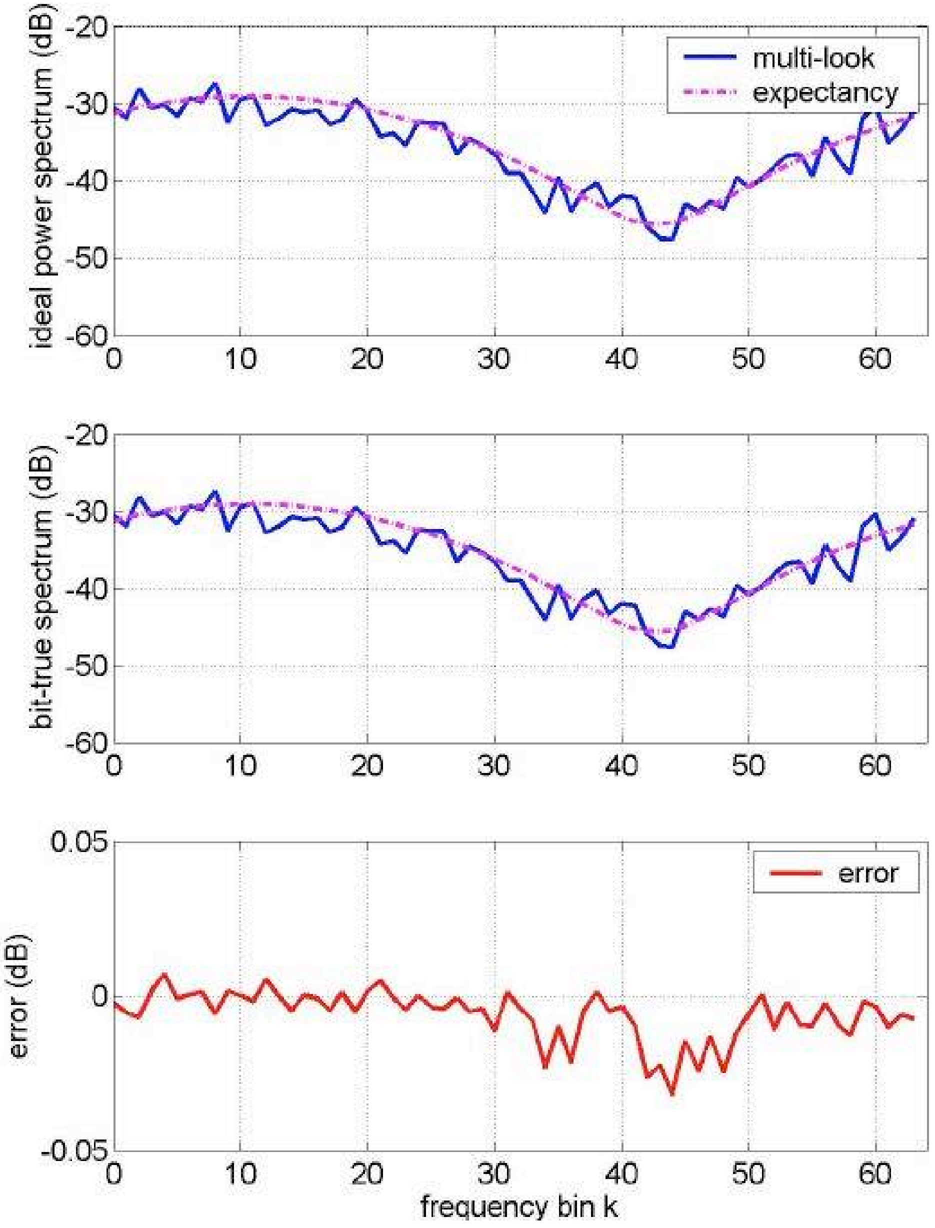
- Purpose: To study how quantization effects like saturation and numeric roundoff impact the quality of the computed Doppler spectra
  → Gauge the impact on the accuracy of rain vertical velocity measurements
- Method for generating realistic rain spectra
  - Use Gaussian-shaped power spectra: Simulates vertical Doppler velocity measurements with a finite gain antenna
  - Center frequency f_0 corresponds to rain's vertical velocity



  - Input data for multiple, independent radar looks (modulate spectrum with random, Rayleigh distributed amplitude and uniformly distributed phase, then inverse FFT)
- Matlab test fixture development
  - Generates time-domain input signal (complex, pulse-compressed radar echoes) for a range of signal levels and f_0 values
  - Compares fixed-point output to that of "golden" (floating-point) FFT model to assess quantization effects

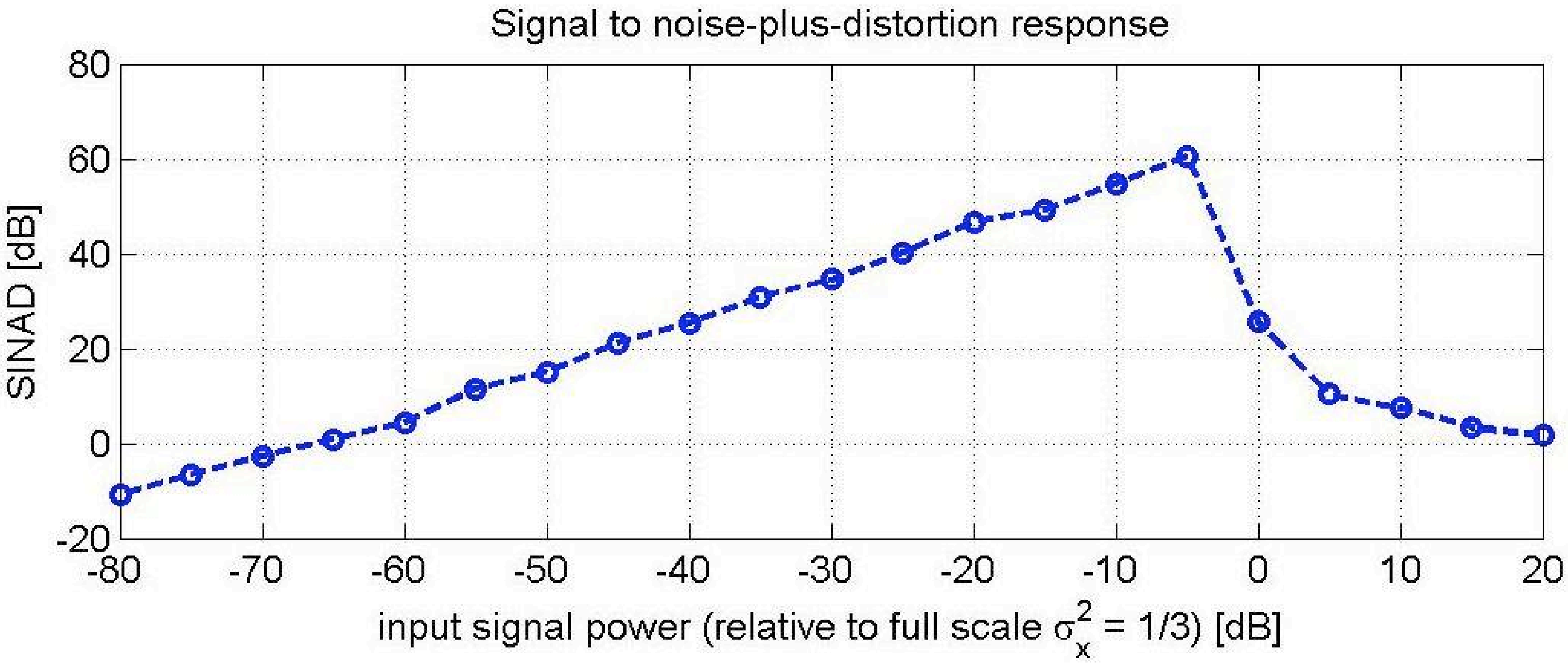
Simulation Results

- Time-domain Doppler rain echo data were generated and input to the fixed-point Simulink design
  - $N_l = 8$ independent radar looks
- At right simulation for a Doppler rain target with a vertical velocity of ~11 m/s ($f_0 = 1$ kHz)
  - Top: the ideal (floating-point) model of the processor core
  - Middle: the bit-true model of the processor core
  - Bottom: the error signal (difference between bit-true and ideal responses)
  - Bias in estimated centroid is 3 Hz or about .03 m/s (very small quantization effect)



Dynamic Range

- Dynamic range was tested by varying the Doppler rain spectra over a range of signal strengths and measuring the output signal-to-noise plus distortion (SINAD) ratio
  - SINAD defined as the ratio of the floating-point signal power out to the noise power out due to fixed-point errors
- Simulations show that a linear region occurs in the response which extends over more than a 60 dB range, in agreement with theory

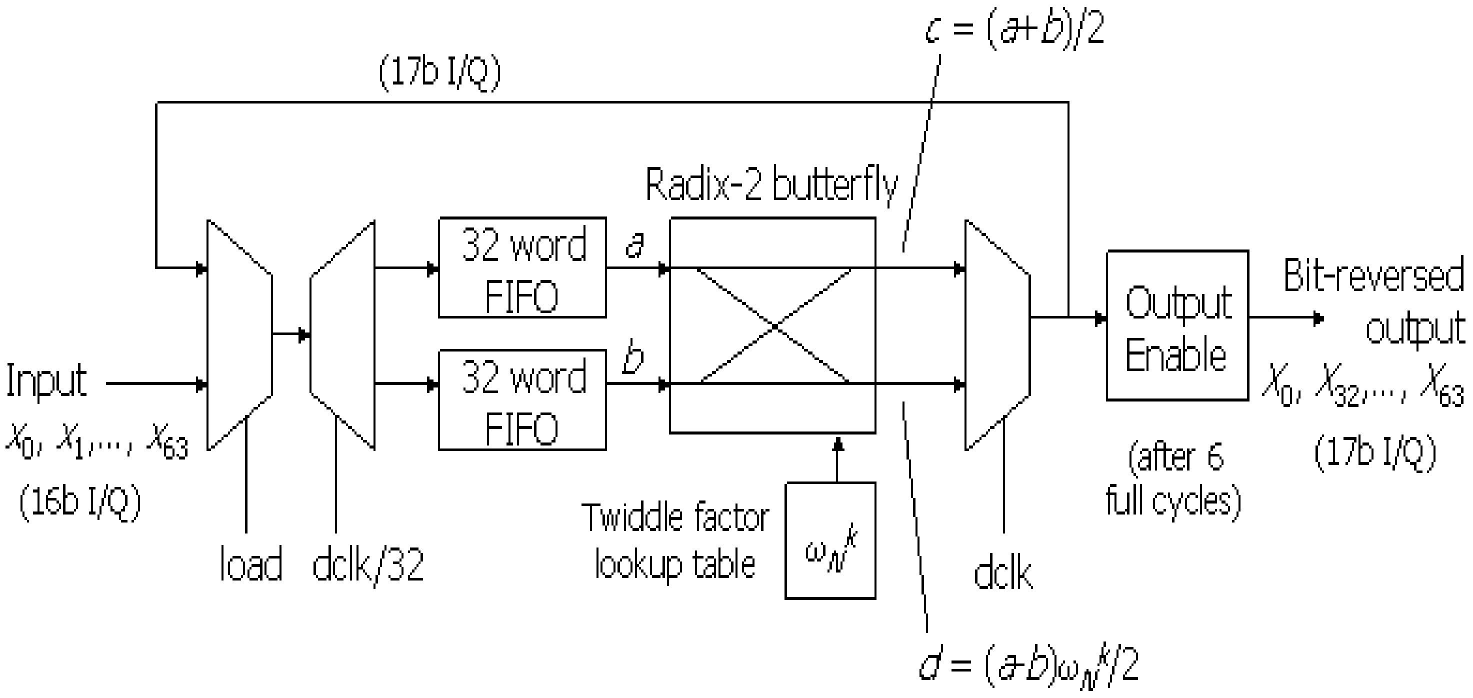


Verilog Design of FFT Processing Block

- The 22 □s execution time limit for one FFT operation could be satisfied by instantiating just one butterfly module and running all 192 data passes
  - to save space this design was initially pursued
- The design is based on the Singleton technique, which uses multiplexer switches to permute the data order into the butterfly and uses two banks of first-in/first-out (FIFO) memory to store intermediate results.
- The twiddle factor coefficients, $□_N^k = e^{-j2□k/N}$, are stored in the FPGA’s lookup table (LUT) and addressed in sequence for the 192 butterfly passes.
- As the final 32 passes are completed, a register is enabled to send the Fourier transformed result out in a data burst to the next processing block.
  - as a consequence of the data permutation, the frequency data is output in a bit-reversed address sequence.
- Single-butterfly FFT core was designed at the top-level in Verilog and used pre-existing IP cores designed by Xilinx at a lower level for generic components such as multipliers.

FFT Core Using Single Butterfly

FPGA hardware architecture implementation of the 64-point FFT core



Expected Performance

- After logic synthesis, the design summary showed that the FFT occupied ~9% of the available configurable logic blocks (CLB) on-chip.
- Behavioral simulations with ModelSim showed that the 64-point transformed result becomes available after 390 system clock cycles (19.5 □s).
- This left too small a margin relative to the 22 □s requirement; design was modified to use two butterflies for factor of two improvement in speed
  - New FFT design still requires only 20% of chip CLBs
- Tests of FFT indicate agreement between Verilog implementation and Simulink bit-true simulations
  - Bit-for-bit agreement for test vectors: zero-lag impulse response, dc signal response, and time-delayed impulse response
  - Rain-like test vectors show some disagreement at the least significant bit level; meets dynamic range requirement as is, but will be resolved

Summary and Plans

- During the first year of this task we have completed the following:
  - high-level radar design and processor requirements
  - processing algorithm design and algorithm simulations to verify requirements
  - preliminary HDL coding of key processing algorithm blocks
- Our near-term plans are:
  - Continue work on testing and debugging of Verilog processor blocks
  - Develop Verilog test fixtures for integration of blocks and perform integrated testing
  - Perform synthesis and functional and timing simulations
  - Generate FPGA bit files
  - Begin development of hardware test setup